

IN THE CLAIMS

The following is a listing of claims of the present application:

1. (Currently Amended) A method for use in a client/server system of reducing interactions between a client and server in association with an application being accessed by the client at the server, the method comprising the steps of:

configuring the server to store a model associated with the application and to execute view-generating and controller logic associated with the application; and

configuring the client to store at least a subset of the model associated with the application and to execute at least a subset of the view-generating and controller logic associated with the application; ;

wherein one or more portions of the application are performed at the client without the client having to interact with the server , and further wherein the client and the server both execute the respective model and view-generating and controller logic resident thereon.

2. (Original) The method of claim 1, wherein the client and server communicate over a HyperText Transport Protocol network.

3. (Original) The method of claim 1, wherein the client performs the one or more portions of the application in accordance with browser software running thereon.

4. (Original) The method of claim 3, wherein the configuring step further comprises the step of partitioning a screen area associated with the browser software into frames.

5. (Original) The method of claim 4, wherein the at least a subset of the model, the view-generating and the controller logic associated with the application are associated with at least one frame and one or more views for display in accordance with the application are associated with at least another frame.

6. (Original) The method of claim 5, wherein the at least one view frame is a visible frame.

7. (Original) The method of claim 5, wherein the at least one frame associated with the at least a subset of the model, the view-generating logic and the controller logic is not a visible frame.

8. (Original) The method of claim 4, wherein the configuring step further comprises forming at least one frame with which application-independent view-generating logic and controller logic is associated.

9. (Original) The method of claim 8, wherein the at least one application-independent view-generating logic and controller logic frame further has an application-independent model associated therewith.

10. (Original) The method of claim 8, wherein the at least one application-independent view-generating logic and controller logic frame serves as an application programming interface for developing views to be displayed in accordance with the application.

11. (Original) The method of claim 10, wherein the views are implemented in accordance with the HyperText Markup Language and the application programming interface is implemented in accordance with the JavaScript language.

12. (Original) The method of claim 1, wherein the at least a subset of the model, the view-generating and the controller logic associated with the application are downloaded from the server to the client upon demand.

13. (Currently Amended) A network-based system, comprising:
a server having at least one processor operative to: (i) store a model associated with an application associated with the server; and (ii) execute view-generating and controller logic associated with the application; and

a client, coupled to the server via a network, having at least one processor operative to: (i) store at least a subset of the model associated with the application; and (ii) execute at least a subset of the view-generating and controller logic associated with the application; ;

wherein one or more portions of the application are performed at the client without the client having to interact with the server such that interactions between the client and server are reduced, and further wherein the client and the server both execute the respective model and view-generating and controller logic resident thereon.

14. (Original) The system of claim 13, wherein the network is a HyperText Transport Protocol network.

15. (Original) The system of claim 13, wherein the client processor performs the one or more portions of the application in accordance with browser software running thereon.

16. (Original) The system of claim 15, wherein the client processor is further operative to partition a screen area associated with the browser software into frames.

17. (Original) The system of claim 16, wherein the at least a subset of the model, the view-generating and the controller logic associated with the application are associated with at least one frame and one or more views for display in accordance with the application are associated with at least another frame.

18. (Original) The system of claim 17, wherein the at least one view frame is a visible frame.

19. (Original) The system of claim 17, wherein the at least one frame associated with the at least a subset of the model, the view-generating logic and the controller logic is not a visible frame.

20. (Original) The system of claim 16, wherein the client processor is further operative to form at least one frame with which application-independent view-generating logic and controller logic is associated.

21. (Original) The system of claim 20, wherein the at least one application-independent view-generating logic and controller logic frame further has an application-independent model associated therewith.

22. (Original) The system of claim 20, wherein the at least one application-independent view-generating logic and controller logic frame serves as an application programming interface for developing views to be displayed in accordance with the application.

23. (Original) The system of claim 22, wherein the views are implemented in accordance with the HyperText Markup Language and the application programming interface is implemented in accordance with the JavaScript language.

24. (Original) The system of claim 13, wherein the at least a subset of the model, the view-generating and the controller logic associated with the application are downloaded from the server to the client upon demand.

25. (Currently Amended) An article of manufacture for use in a client/server system for reducing interactions between a client and server in association with an application being accessed by the client at the server, comprising machine readable media containing one or more programs which when executed implement the steps of:

configuring the server to store a model associated with the application and to execute view-generating and controller logic associated with the application; and

configuring the client to store at least a subset of the model associated with the application and to execute at least a subset of the view-generating and controller logic associated with the application; ;

wherein one or more portions of the application are performed at the client without the client having to interact with the server, and further wherein the client and the server both execute the respective model and view-generating and controller logic resident thereon.

26. (Currently Amended) A method for use in a client/server system of reducing interactions between a client and server in association with an application being accessed by the client at the server, the method comprising the steps of:

configuring the server to: (i) store a model associated with the application; (ii) execute view-generating logic associated with the application; and (iii) execute controller logic associated with the application; and

configuring the client to: (i) store at least a subset of the model associated with the application; (ii) execute at least a subset of the view-generating logic associated with the application; and (iii) execute at least a subset of the controller logic associated with the application;

wherein one or more portions of the application are performed at the client without the client having to interact with the server, and further wherein the client and the server both execute the respective model and view-generating and controller logic resident thereon;

further wherein, in accordance with such a dual model-view-controller arrangement, a model comprises application data, rules, and algorithms affecting the data, a view comprises a screen or window representation of a subset of the model that the application chooses to display, and a controller comprises the logic that processes user requests, and causes the model to be changed and the view to be refreshed.